Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	0	(command adj line adj interface) same ((generat$5 produc$5) adj (code script) same (inverse adj handler) same (wrapper adj routine) same (configuration adj dump))	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/11/03 09:48
L2	0	345/810.ccls. and (command adj line adj interface) same ((generat$5 produc$5) adj (code script) same (inverse adj handler) same (wrapper adj routine) same (configuration adj dump))	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/11/03 ~~09:51~~
L3	0	703/21,23.ccls. and (command adj line adj interface) same ((generat$5 produc$5) adj (code script) same (inverse adj handler) same (wrapper adj routine) same (configuration adj dump))	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/11/03 09:51
L4	0	707/500.ccls. and (command adj line adj interface) same ((generat$5 produc$5) adj (code script) same (inverse adj handler) same (wrapper adj routine) same (configuration adj dump))	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/11/03 09:52
L5	0	370/338,229.ccls. and (command adj line adj interface) same ((generat$5 produc$5) adj (code script) same (inverse adj handler) same (wrapper adj routine) same (configuration adj dump))	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/11/03 09:53
L6	0	345/810.ccls. and ((configur$4) near10 (network adj devic$2)) and forms and (script$4 adj language)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/11/03 09:52
L8	1	703/21,23.ccls. and ((configur$4) near10 (network adj devic$2)) and forms and (script$4 adj language)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/11/03 09:51
L9	0	707/500.ccls. and ((configur$4) near10 (network adj devic$2)) and forms and (script$4 adj language)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/11/03 09:52

L10	0	345/810.ccls. and ((configur$4) near10 (network adj devic$2)) and forms and (script$4 adj language)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/11/03 09:52
L11	1	370/338,229.ccls. and ((configur$4) near10 (network adj devic$2)) and forms and (script$4 adj language)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/11/03 09:53
S1	24	((configur$4) near10 (network adj devic$2)) and forms and (script$4 adj language)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/08/19 10:55
S2	418283	(((configur$4) near10 (network adj devic$2)) and forms and (script$4 adj language)) and 709/220, "221", "223", "224".ccls.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/03/22 14:01
S3	6297	709/220,221,223,224.ccls.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/03/22 16:59
S4	5	709/220,221,223,224.ccls. and (((configur$4) near10 (network adj devic$2)) and forms and (script$4 adj language))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/08/19 13:08
S5	704	((configur$4) near2 (network adj device))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/03/23 11:59
S6	258	((obtain$4 receiv$4) near2 (network adj device)) near2 (state information configur$5)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/03/23 11:59
S7	106	(((obtain$4 receiv$4) near2 (network adj device)) near2 (state information configur$5)) and (((configur$4) near2 (network adj device)))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/03/22 14:09

S8	33	((((obtain$4 receiv$4) near2 (network adj device)) near2 (state information configur$5)) and (((configur$4) near2 (network adj device)))) and ((user operator) adj interface)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/03/22 14:09
S9	68	((obtain$4) near2 (network adj device)) near2 (state information configur$5)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/03/22 17:00
S10	28	709/220,221,223,224.ccls. and((obtain$4) near2 (network adj device)) near2 (state information configur$5)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/03/22 16:00
S11	0	709/220,221,223,224.ccls. and (button field window) and (requestquerystring)and (requestform)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/03/22 16:09
S12	0	709/220,221,223,224.ccls. and (requestquerystring)and (requestform)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/03/22 16:09
S13	0	(requestquerystring)and (requestform)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/03/22 16:15
S14	494495	(request query string)and (request form)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/03/22 16:10
S15	5245	((request query string)and (request form)) and 709/220,221,223,224.ccls.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/03/22 16:10
S16	27	(((request query string)and (request form)) and 709/220,221,223,224.ccls.) and (((obtain$4) near2 (network adj device)) near2 (state information configur$5))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/03/22 16:10
S17	1	("request.querystring")and ("request.form")	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/03/22 16:59

S18	1361	709/220,221,223,224.ccls. and (GUI (graphical adj user adj interface))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/03/22 17:00
S19	7	(709/220,221,223,224.ccls. and (GUI (graphical adj user adj interface))) and ((obtain$4) near2 (network adj device)) near2 (state information configur$5)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/03/22 17:00
S20	52	(709/220,221,223,224.ccls. and (GUI (graphical adj user adj interface))) and((configur$4) near2 (network adj device))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/03/23 10:52
S21	0	"09223565" and lindhorst	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/03/23 14:07
S22	0	((BROWSER FORM)NEAR2 (BASED DRIVEN DEPENDENT)) NEAR2 (NETWORK ADJ (MANAG$4))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/03/23 11:26
S23	8144	BROWSER FORM)ADJ(BASED DRIVEN DEPENDENT	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/03/23 11:26
S24	1	(BROWSER FORM)ADJ(BASED DRIVEN DEPENDENT) NEAR5 (NETWORK ADJ (MANAG$4))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/03/23 11:59
S25	1	((BROWSER FORM)ADJ(BASED DRIVEN DEPENDENT) and (NETWORK ADJ (MANAG$4))) and ((configur$4) near2 (network adj device))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/03/23 11:59
S26	2	((BROWSER FORM)ADJ(BASED DRIVEN DEPENDENT) and (NETWORK ADJ (MANAG$4))) and ((obtain$4 receiv$4) near2 (network adj device)) near2 (state information configur$5)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/03/23 12:14
S27	2	"5774667".pn.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/03/23 12:15

S28	2	"6308205".pn.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/03/23 12:15
S29	1	(09/223565) and lindhorst	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/03/23 14:08
S30	2	(09/264874) and (sugiyama)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/08/19 10:58
S31	35	ASP adj web adj Page$5	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/08/19 12:27
S32	2	((one single unit) near3 (browser form) near10 (Obtain$5 receiv$6 get$5)near10 (state) near10 (devic$5 applianc$5)) and ((chang$5 alter$5 configur$5) near5 (state) near10 (devic$5 applianc$5))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/08/19 13:41
S33	0	((one single unit) near3 (browser) near10 (Obtain$5 receiv$6 get$5)near10 (state) near10 (devic$5 applianc$5)) and ((chang$5 alter$5 configur$5) near5 (state) near10 (devic$5 applianc$5))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/08/19 12:35
S34	0	((one single unit) near10(browser) near10 (Obtain$5 receiv$6 get$5)near10 (state) near10 (devic$5 applianc$5)) and ((chang$5 alter$5 configur$5) near5 (state) near10 (devic$5 applianc$5))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/08/19 12:35
S35	0	((one singl$5) near10(browser) near10 (Obtain$5 receiv$6 get$5)near10 (state) near10 (devic$5 applianc$5)) and ((chang$5 alter$5 configur$5) near5 (state) near10 (devic$5 applianc$5))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/08/19 12:36

S36	5	((one singl$5) near10(browser) near10 (Obtain$5 receiv$6 get$5)near10 (state status configur$6) near10 (devic$5 applianc$5)) and ((chang$5 alter$5 configur$5) near5 (state status configur$6) near10 (devic$5 applianc$5))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/08/19 12:41
S37	33	((browser) near10 (Obtain$5 receiv$6 get$5)near10 (state status configur$6) near10 (devic$5 applianc$5)) and ((chang$5 alter$5 configur$5) near5 (state status configur$6) near10 (devic$5 applianc$5))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/08/19 12:42
S38	0	(09/746673) and chunguang	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/08/19 13:11
S39	0	(09/746673) and singh	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/08/19 13:11
S40	0	("09746673") and gurminder	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/08/19 13:11
S41	0	("09/746673") and gurminder	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/08/19 13:12
S42	0	((BROWSER FORM)ADJ(BASED DRIVEN DEPENDENT) and (NETWORK ADJ (MANAG$4))) and (submit) near10 (execut$6) near10 (routin$6)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/08/19 13:37
S43	0	((BROWSER FORM)ADJ(BASED DRIVEN DEPENDENT) and (NETWORK ADJ (MANAG$4))) and (submit$5) near10 (execut$6) near10 (routin$6)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/08/19 13:15
S44	20	((BROWSER FORM)ADJ(BASED DRIVEN DEPENDENT) and (NETWORK ADJ (MANAG$4))) and (execut$6) near10 (routin$6)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/08/19 13:15

S45	270	(BROWSER FORM)ADJ(BASED DRIVEN DEPENDENT) and (NETWORK ADJ (MANAG$4))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/08/19 13:37
S46	203	(BROWSER)ADJ(BASED DRIVEN DEPENDENT) and (NETWORK ADJ (MANAG$4))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/08/19 13:19
S47	108	(FORM)ADJ(BASED DRIVEN DEPENDENT) and (NETWORK ADJ (MANAG$4))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/08/19 13:19
S48	0	(FORM)ADJ(BASED DRIVEN DEPENDENT) near15 (NETWORK ADJ (MANAG$4))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/08/19 13:19
S49	1	(BROWSER)ADJ(BASED DRIVEN DEPENDENT) near15 (NETWORK ADJ (MANAG$4))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/08/19 13:21
S50	2	(associat$5 excut$5 connect$5) near10 (routin$5 program$5) near10 (submit adj button)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/08/19 13:23
S51	11	(associat$5 excut$5 connect$5) near10 (routin$5 program$5 method$5) near10 (submit adj button)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/08/19 13:31
S52	10	(associat$5 excut$5) near10 (routin$5 program$5 method$5) near10 (submit adj button)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/08/19 13:32
S53	12	(associat$5 excut$5) near15 (routin$5 program$5 method$5) near15 (submit adj button)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/08/19 13:32
S54	0	((web)ADJ(BASED DRIVEN DEPENDENT) and (NETWORK ADJ (MANAG$4))) and (submit) near10 (execut$6) near10 (routin$6)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/08/19 13:37

S55	16	(web)ADJ(BASED DRIVEN DEPENDENT) near15 (NETWORK ADJ (MANAG$4))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/08/19 13:37
S56	0	(((web)ADJ(based)) near10 (Obtain$5 receiv$6 get$5)near10 (state) near10 (devic$5 applianc$5)) and ((chang$5 alter$5 configur$5) near5 (state) near10 (devic$5 applianc$5))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/08/19 13:48
S57	0	(((web)near5(based dependent$5)) near10 (Obtain$5 receiv$6 get$5)near10 (state) near10 (devic$5 applianc$5)) and ((chang$5 alter$5 configur$5) near5 (state) near10 (devic$5 applianc$5))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/08/19 13:48
S58	4	((web-based) near10 (network near5 manag$5)) and ((Obtain$5 receiv$6 get$5)near10 (state) near10 (devic$5 applianc$5)) and ((chang$5 alter$5 configur$5) near5 (state) near10 (devic$5 applianc$5))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/08/19 13:58
S59	0	((web-based) near10 (network near5 devic$5 near10 manag$5)) and ((Obtain$5 receiv$6 get$5)near10 (state) near10 (devic$5 applianc$5)) and ((chang$5 alter$5 configur$5) near5 (state) near10 (devic$5 applianc$5))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/08/19 16:18
S60	2	"6308205".pn.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/08/19 16:18
S61	11	(associat$5 excut$5 connect$5) near10 (routin$5 program$5 method) near10 (submit adj button)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/08/20 08:39
S62	1	((ASP script) near3 (statement$5 data file (data adj structure))) near10 (Obtain$5 receiv$6 get$5)near10 (state) near10 (devic$5 applianc$5) and ((chang$5 alter$5 configur$5) near5 (state) near10 (devic$5 applianc$5))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/11/02 11:59

S63	1530	command adj line adj interface	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/11/02 11:59
S64	17	(command adj line adj interface) and (Obtain$5 receiv$6 get$5)near10 (state) near10 (devic$5 applianc$5) and ((chang$5 alter$5 configur$5) near5 (state) near10 (devic$5 applianc$5))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/11/02 12:35
S65	2	(command adj line adj interface) and ((generat$5 produc$5 provid$5) near5 (code script) same (inverse adj handler) same (wrapper adj routine) same (configuration adj dump))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/11/03 09:09
S66	2	(command adj line adj interface) and ((generat$5 produc$5 provid$5) near5 (code script) same (inverse adj handler) and (wrapper adj routine) same (configuration adj dump))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/11/02 12:28
S67	2	(command adj line adj interface) and ((generat$5 produc$5 provid$5) near5 (code script) same (inverse adj handler) and (wrapper adj routine) and (configuration adj dump))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/11/02 12:29
S68	2	(command adj line adj interface) and (generat$5 produc$5 provid$5) near5 (code script) and (inverse adj handler) and (wrapper adj routine) and (configuration adj dump)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/11/02 12:29
S69	2	(command adj line adj interface) and (generat$5 produc$5 provid$5) near5 (code script) and (configuration adj dump)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/11/02 12:30
S70	2	(command adj line adj interface) and (generat$5 produc$5 provid$5) near5 (code script command$5) and (configuration adj dump)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/11/02 12:30
S71	719	(command adj line adj interface) and (generat$5 produc$5 provid$5) near5 (code script command$5)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/11/02 12:30

S72	11	(command adj line adj interface) and ((generat$5 produc$5 provid$5) near5 (code script command$5) and (Obtain$5 receiv$6 get$5)near10 (state) near10 (devic$5 applianc$5) and ((chang$5 alter$5 configur$5) near5 (state) near10 (devic$5 applianc$5)))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/11/02 12:36
S73	2	((generat$5 produc$5 provid$5) near5 (code script) same (inverse adj handler) same (wrapper adj routine) same (configuration adj dump))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/11/02 12:36
S74	649	((generat$5 produc$5 provid$5) near5 (code script command$5) and (Obtain$5 receiv$6 get$5)near10 (state) near10 (devic$5 applianc$5) and ((chang$5 alter$5 configur$5) near5 (state) near10 (devic$5 applianc$5)))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/11/02 14:45
S75	8	((generat$5 produc$5 provid$5) near5 (code script command$5) near10 (Obtain$5 receiv$6 get$5)near10 (state) near10 (devic$5 applianc$5) near10 ((chang$5 alter$5 configur$5) near5 (state) near10 (devic$5 applianc$5)))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/11/02 12:41
S76	549	(Form (submit adj button)) and ((generat$5 produc$5 provid$5) near5 (code script command$5) and (Obtain$5 receiv$6 get$5)near10 (state) near10 (devic$5 applianc$5) and ((chang$5 alter$5 configur$5) near5 (state) near10 (devic$5 applianc$5)))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/11/02 12:48
S77	549	((Form (submit adj button)) and ((generat$5 produc$5 provid$5)) near5 (code script command$5) and (Obtain$5 receiv$6 get$5)near10 (state) near10 (devic$5 applianc$5) and ((chang$5 alter$5 configur$5) near5 (state) near10 (devic$5 applianc$5)))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/11/02 13:13
S78	3	((Form and (submit adj button)) and ((generat$5 produc$5 provid$5)) near5 (code script command$5) and (Obtain$5 receiv$6 get$5)near10 (state) near10 (devic$5 applianc$5) and ((chang$5 alter$5 configur$5) near5 (state) near10 (devic$5 applianc$5)))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/11/02 13:27

S79	1	((Form near10 (submit adj button)) and ((generat$5 produc$5 provid$5)) near5 (code script command$5) and (Obtain$5 receiv$6 get$5)near10 (state) near10 (devic$5 applianc$5) and ((chang$5 alter$5 configur$5) near5 (state) near10 (devic$5 applianc$5)))	US-PGPUB;
			USPAT;
			EPO; JPO;
			DERWENT;
			IBM_TDB
				OR	ON	2004/11/02 13:37

S80	3	(((submit adj button)) and ((generat$5 produc$5 provid$5)) near5 (code script command$5) and (Obtain$5 receiv$6 get$5)near10 (state) near10 (devic$5 applianc$5) and ((chang$5 alter$5 configur$5) near5 (state) near10 (devic$5 applianc$5)))	US-PGPUB;
			USPAT;
			EPO; JPO;
			DERWENT;
			IBM_TDB
				OR	ON	2004/11/02 13:38

S81	27	709/220,221,223,224.ccls. and((generat$5 produc$5 provid$5) near5 (code script command$5) and (Obtain$5 receiv$6 get$5)near10 (state) near10 (devic$5 applianc$5) and ((chang$5 alter$5 configur$5) near5 (state) near10 (devic$5 applianc$5)))	US-PGPUB;
			USPAT;
			EPO; JPO;
			DERWENT;
			IBM_TDB
				OR	ON	2004/11/02 14:48

IEEE HOME | SEARCH IEEE | SHOP | WEB ACCOUNT | CONTACT IEEE

IEEE

Membership Publications/Services Standards Conferences Careers/Jobs

IEEE Xplore®
RELEASE 1.8

Welcome
United States Patent and Trademark Office

» Se

Help FAQ Terms IEEE Peer Review

Quick Links

Welcome to IEEE Xplore®

- ○- Home
- ○- What Can I Access?
- ○- Log-out

Tables of Contents

- ○- Journals & Magazines
- ○- Conference Proceedings
- ○- Standards

Search

- ○- By Author
- ○- Basic
- ○- Advanced
- ○- CrossRef

Member Services

- ○- Join IEEE
- ○- Establish IEEE Web Account
- ○- Access the IEEE Member Digital Library

IEEE Enterprise

- ○- Access the IEEE Enterprise File Cabinet

Your search matched **0** of **1088345** documents.
A maximum of **500** results are displayed, **15** to a page, sorted by **Relevance Descending** order.

Refine This Search:
You may refine your search by editing the current search expression or enteri new one in the text box.

form<and>script<and>device configuration Search

□ Check to search within this result set

Results Key:
JNL = Journal or Magazine **CNF** = Conference **STD** = Standard

Results:
No documents matched your query.

Print Format

Home | Log-out | Journals | Conference Proceedings | Standards | Search by Author | Basic Search | Advanced Search | Join IEEE | Web Account | New this week | OPAC Linking Information | Your Feedback | Technical Support | Email Alerting | No Robots Please | Release Notes | IEEE Online Publications | Help | FAQ| Terms | Back to Top

Subscribe (Full Service) Register (Limited Service, Free) Login

Search: ● The ACM Digital Library ○ The Guide

+Submit +button +device +configuration ASP script SEARCH

THE ACM DIGITAL LIBRARY

Feedback Report a problem Satisfaction survey

Published before November 2000
Terms used **Submit button device configuration ASP script**

Found **5** of **107,388**

Sort results by: relevance
Display results: expanded form

Save results to a Binder
Search Tips
□ Open results in a new window

Try an Advanced Search
Try this search in The ACM Guide

Results 1 - 5 of 5

Relevance scale

1 Writing CGI Scripts in Python
Michel Vanaken
February 1997 **Linux Journal**
Full text available: html(22.55 KB) Additional Information: full citation, references, index terms

2 SNMP through WWW
Ching-Wun 'Bo' Tsai, Ruay-Shiung 'Bo' Chang
March 1998 **International Journal of Network Management**, Volume 8 Issue 2
Full text available: pdf(376.25 KB) Additional Information: full citation, abstract, references, citings, index terms

In this article we propose a bilingual agent to accept either SNMP or HTTP commands and design several HTML pages to facilitate the task of network management. For network elements that support only SNMP, the bilingual agent can act as a proxy, so that the traditional SNMP agent can also be queried through the Web browser. © 1998 John Wiley & Sons, Ltd.

3 MultiFax
Marcel Gagne
December 1999 **Linux Journal**
Full text available: html(21.03 KB) Additional Information: full citation, abstract, references, index terms

Psst! Want to create a Windows broadcast fax system with web-based administration using Linux? Come over here and we'll talk

4 Haptic rendering: programming touch interaction with virtual objects
K. Salisbury, D. Brock, T. Massie, N. Swarup, C. Zilles
April 1995 **Proceedings of the 1995 symposium on Interactive 3D graphics**
Full text available: pdf(897.60 KB) Additional Information: full citation, abstract, references, citings, index terms

Haptic rendering is the process of computing and generating forces in response to user interactions with virtual objects. Recent efforts by our team at MIT's AI laboratory have resulted in the development of haptic interface devices and algorithms for generating the forces of interaction with virtual objects. This paper focuses on the software techniques needed to generate sensations of contact interaction and material properties. In particular, the techniques we describe are appropriate fo ...

5 Distributed environment: Nestmod: the NetMod - NEST interface
Utpal Amin, David W. Bachmann, Kerman Deboo, Toby J. Teorey
October 1991 **Proceedings of the 1991 conference of the Centre for Advanced Studies on Collaborative research**
Full text available: pdf(1.02 MB) Additional Information: full citation, abstract, references, citings

NESTMOD is a combined analytical modeling and simulation tool, based on the existing tools NetMod and NEST. It provides both transient and steady-state response statistics from models of interconnected local area networks that can execute at any level of detail desired. This gives users the potential to model both networks of extremely large scope (hundreds of thousands of nodes), and to look at great detail for any combination of the ISO layers. This paper describes the interface implementation ...

Results 1 - 5 of 5

PORTAL
US Patent & Trademark Office

Subscribe (Full Service) Register (Limited Service, Free) Login

Search: ● The ACM Digital Library ○ The Guide

+Form +Submit +button +device +configuration ASP script SEARCH

THE ACM DIGITAL LIBRARY

Feedback Report a problem Satisfaction survey

Published before November 2000
Terms used
Form Submit button device configuration ASP script

Found **1** of **107,388**

Sort results by: relevance
Display results: expanded form

Save results to a Binder
Search Tips
□ Open results in a new window

Try an Advanced Search
Try this search in The ACM Guide

Results 1 - 1 of 1

Relevance scale

1 Writing CGI Scripts in Python
Michel Vanaken
February 1997 **Linux Journal**

Full text available: html(22.55 KB)

Additional Information: full citation, references, index terms

Results 1 - 1 of 1

PORTAL

US Patent & Trademark Office

Subscribe (Full Service) Register (Limited Service, Free) Login

Search: ● The ACM Digital Library ○ The Guide

+Form +Submit +button +device +configuration ASP script SEARCH

THE ACM DIGITAL LIBRARY

Feedback Report a problem Satisfaction survey

Published before November 2000
Terms used
Form Submit button device configuration ASP script

Found **1** of **107,388**

Sort results by: relevance
Display results: expanded form

Save results to a Binder
Search Tips
□ Open results in a new window

Try an Advanced Search
Try this search in The ACM Guide

Results 1 - 1 of 1

Relevance scale

1 Writing CGI Scripts in Python
Michel Vanaken
February 1997 **Linux Journal**

Full text available: html(22.55 KB)

Additional Information: full citation, references, index terms

Results 1 - 1 of 1

Terms of Usage Privacy Policy Code of Ethics Contact Us

Useful downloads: Adobe Acrobat QuickTime Windows Media Player Real Player

A Java Based XML Browser for Consumer Devices

Petri Vuorimaa, Teemu Ropponen, Niklas von Knorring, and Mikko Honkala
Telecommunications Software and Multimedia Laboratory
Helsinki University of Technology
P.O. Box 5400, FI-02015 HUT, Finland
Tel. +358-9-451 4794, Fax +358-9-451 5253
E-mail: Petri.Vuorimaa@hut.fi

ABSTRACT

Next generation consumer devices will all have an Internet connection. Thus, one vision is that the future multimedia services will be browser based. Extensible Markup Language (XML) is the most likely markup language. In this paper, we introduce a Java based XML browser called X-Smiles. It is intended for consumer devices and supports multimedia services. The main advantage of the X-Smiles browser is that it supports most of the XML related specifications. Different XML based languages can be mixed freely in applications. In addition, the X-Smiles has special user interfaces for different kinds of devices (e.g., digital television, personal digital assistants, and mobile phones). These user interfaces can be used as so called virtual prototypes of the real devices. The X-Smiles browser is available as open source at http://www.x-smiles.org.

Categories and Subject Descriptors

H.5.4 **[Information Interfaces and Presentation (e.g., HCI)]**: Hypertext/Hypermedia – *architectures, navigation, user issues.*

General Terms

Design, Human Factors, Standardization, Languages

Keywords

XML, XSL FO, SVG, SMIL, Multimedia

1. INTRODUCTION

Next generation computers will be non-desktop devices. This means that different kinds of mobile devices (e.g., personal digital assistants (PDAs), smart phones, and web pads) and digital television Set-Top Boxes (STBs) will outnumber the personal computers connected to the Internet. The difference between these new devices and personal computers is that they come with varying sizes and properties. For example, different kinds of mobile devices have very different screen sizes and input mechanisms [14].

The uniting factor between the different appliances is that they are all connected to the Internet. This will mean that, although, there is divergence on the network and hardware level, there can be convergence on the software system level. Since most of the Internet services are currently browser based, the future multimedia services will also likely be such.

We have been studying the software architecture of mobile devices in the GO-MM project (http://go.cs.hut.fi) [19]. The architecture is based on Linux operating system, which as such is too complicated for embedded devices. Therefore, we have integrated the Transvirtual Technologies Kaffe Java Virtual Machine directly on top of a Vesa frame buffer graphics library [16]. The main advantage of this approach is that unnecessary parts (e.g., X-windows, network file system, and virtual memory) can be removed from the Linux. The network connection is based on the IEEE802.11 wireless Ethernet [9] and Internet multimedia protocols [13].

The major part of the project has been the development of an XML [4] browser called X-Smiles. The main advantage of the XML is that it allows the versioning of the multimedia services into different kinds of end-user devices [3], [15]. The content of the services is coded using the XML, while the user interface is defined in a separate eXtensible Stylesheet Language (XSL) [1] file.

There are also other XML based browsers available. For example, *InDelv*, *Microsoft Internet Explorer*, *Netscape Mozilla*, and *World Wide Web Consortium (W3C) Amaya* browsers have XML support. They do not, however, support all the XML related specifications. The main reason for this is that all the above-mentioned browsers, except InDelv, are originally designed for HTML. Thus, integrating new XML features is difficult and makes the browsers bulky.

The main advantage of the X-Smiles browser is that it supports several XML related specifications and is still suitable for embedded devices supporting Java. The X-Smiles browser supports, e.g., XSL Formatting Objects (XSL FO) [1], Synchronized Multimedia Integration Language (SMIL) [12], and Scalable Vector Graphics (SVG) [7]. These are all specification made by the W3C.

The XSL FO is a vocabulary of graphic primitives, which are used to define the layout of an XML document. XSL Transformations (XSLT), on the other hand, is an XML language used to restructure, or transform the original document. Together, XSLT and XSL FO comprise XSL, a language for defining stylesheets. XSL provides a powerful means for layout, which extends the capabilities of, e.g., Cascading StyleSheets (CSS), and gives the developer more control over the look of the document.

SAC 2002, Madrid, Spain
© 2002 ACM 1-58113-445-2/02/3...$5.00.

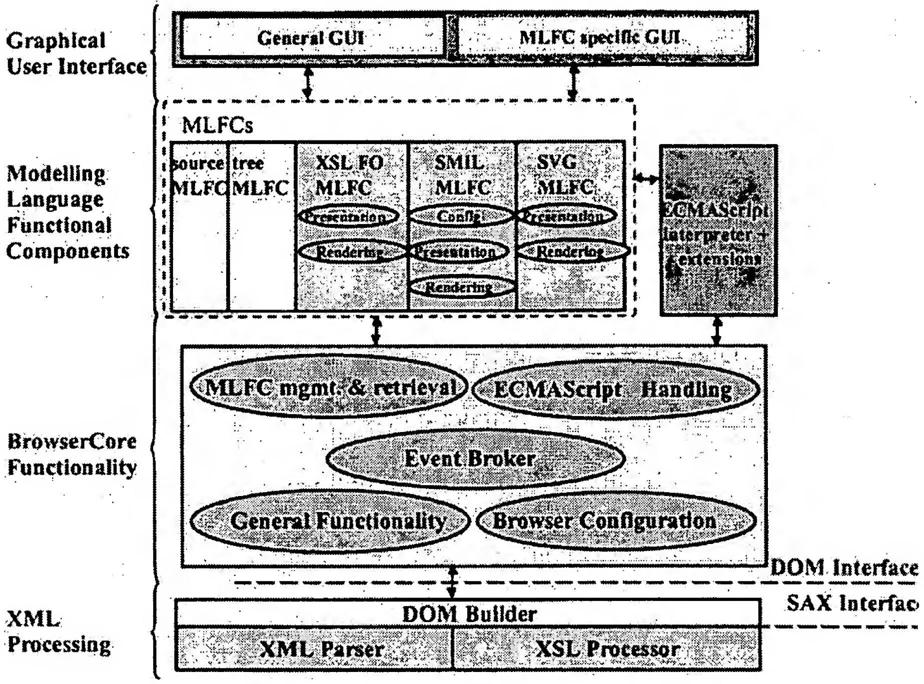


Figure 1. The architecture of the X-Smiles browser.

SMIL is an XML based language, which can be used, e.g., to define the spatial and temporal relationships between the media elements in order to create multimedia presentation. SMIL can be used for streaming presentations containing all kinds of multimedia elements – not just audio and video.

Finally, SVG is an XML language for describing vector graphics. In many cases, vector graphics is a more precise and compact way to present graphics than bitmap images. With SVG it is possible to embed graphics – and even animations – on www pages. The user can also control (e.g., zoom and pan) the presentation of the graphics.

In the following, we will first present the architecture of the X-Smiles browser. After that, its operation is discussed. Then, service examples on the X-Smiles browser are shown. Finally, conclusions are given.

2. X-SMILES ARCHITECTURE

The architecture of the X-Smiles browser is depicted in Figure 1. The architecture is composed of five major layers (from bottom to the top): *XML processing*, *Browser Core Functionality*, *Modeling Language Functional Components (MLFCs)* & *ECMAScript Interpreter*, and *Graphical User Interfaces (GUIs)*. Each layer is described in the following.

The XML processing module contains the *XML Parser* and *XSL Processor*. The XML Parser parses the XML documents. The XSL Processor processes the document according to the related XSL stylesheet, if applicable. The output is stored into a data structure, which can be accessed via the *Document Object Model (DOM)* [2] *Interface*.

The core of the browser ties the different modules together. The main parts of the core are *MLFC Management and Retrieval*, *ECMAScript Handler*, *Event Broker*, *General Functionality*, and *Browser Configuration*. The MLFC Management and Retrieval unit takes care of loading the appropriate MLFC, which can also be retrieved over the network. The Event Broker forwards the events to other units. The ECMAScript Handler co-ordinates the operation of the ECMAScript interpreter. The General Functionality unit controls the GUI, document history, etc. Finally, the Browser Configuration unit is responsible for management of the different features of the browser such as the parser and stylesheet processor, home page, etc.

The processed documents can be displayed by different MLFCs. They allows the easy integration of different kinds of XML languages to the X-Smiles browser. The different XML languages can be integrated as separate components, which can loaded dynamically – even on the fly. So far, we have implemented MLFCs for source code, document element tree, XSL FO, SMIL, and SVG. The different MLFCs are discussed in detail in the next Section.

The *ECMAScript* [6] *Interpreter* runs the scripts contained in the XSL stylesheet. ECMAScript provides service developers a convenient method to implement interactivity. The scripts are defined in CDATA sections embedded in the XML document or in the XSL stylesheet. The scripts are run either when documents are loaded or in response to user or timer events. For example, a user action, such as clicking a mouse button on an image, might create an event, which in turn activates a function defined using ECMAScript.

The original X-Smiles GUI was developed for desktop PCs. We have also developed a special user interface for digital television STBs [18], PDAs, and mobile phones. The different GUIs are interchangeable, and the user can switch them even on the fly. The main advantage of this feature is that XML service developer can easily test the content in different kinds of devices. The different GUIs and their implementation are discussed in detail in the following Section.

3. X-SMILES OPERATION

In this Section, we review in more detail the operation of the X-Smiles browser. First, we describe XML parsing and XSL processing. Next, different MLFCs and GUIs are introduced in detail. Finally, different implementation issues are discussed.

3.1 XML Parsing and XSL Processing

When the X-Smiles browser reads a new XML document, it has to do certain steps to process it. First, the browser reads the document source. The file is accessed using either the *file* or *http* protocol. Secured connections (i.e., *https*) are not supported, yet. The processing is done using any XML parser, which uses the Simple API for XML (SAX), such as the *Apache Xerces* parser (http://xml.apache.org/xerces-j/). The parser processes the XML file line by line and creates different SAX events. The DOM implementation works together with the parser. It takes the SAX events as input and constructs the DOM model of the document accordingly.

Next, the XSL stylesheet is read. First, the browser determines, whether a stylesheet is attached to the XML document. If a stylesheet is associated with the XML file, the browser reads the stylesheet and creates a DOM data structure. Since the XSL stylesheet is also a valid XML document, it can be processed in the same way as the original XML document.

The next step is to actually process the stylesheet. The browser calls the XSL processor with the stylesheet and document as parameters. Both are DOM trees as described above. The XSL processor creates a new DOM tree from the XML document based on the instructions given in the XSL stylesheet. Currently, we use the *Apache Xalan* XSL processor (http://xml.apache.org/xalan/).

3.2 Markup Language Functional Components (MLFCs)

After the final DOM tree has been constructed, it has to be rendered. Since there are different XML languages (e.g., MathML, ChemicalML, SMIL, XSL FO, and SVG) available, different rendering modules have to be used. In X-Smiles, this is achieved by using different MLFCs. Each MLFC knows how to render one specific type of XML language. The different MLFCs can be used at the same time. For example, XSL FO document can contain a SMIL document. So far, we have implemented a MLFC for SMIL, XML FO, and SVG.

In addition, we have basic MLFCs, which can display source code and tree view. The source code MLFC is useful for debugging. The content of the original XML document, an associated XSL stylesheet, as well as the resulting source code of the processed document can be viewed. The tree view presents the DOM structure of the processed document in a tree-like format, thus providing an easier way to examine the structure of the document.

The SMIL MLFC was the first true MLFC that was developed. The SMIL MLFC supports most of the features of the SMIL version 1.0. SMIL 2.0 Basic Profile is also supported. The Java Media Framework (JMF) [10], extended with custom JMF players developed in the project, is used for rendering and synchronization of both continuous (e.g., audio and video) and static (e.g., text and images) media elements. An example of a SMIL document rendered by the SMIL MLFC is shown in Figure 2.

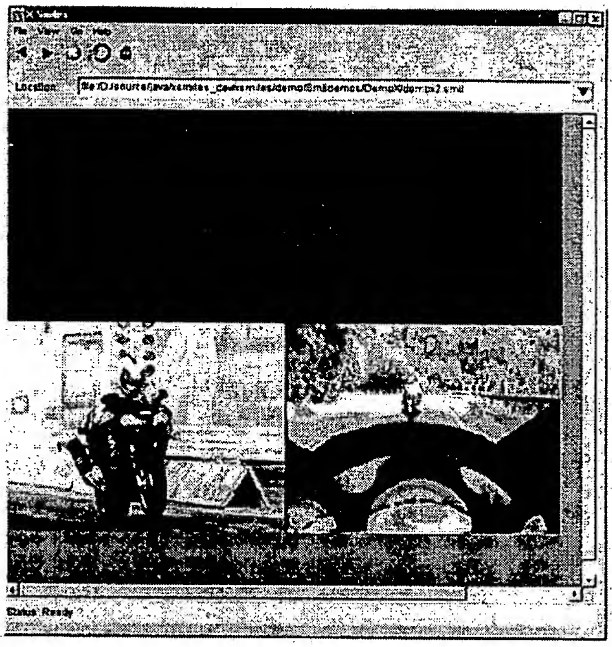
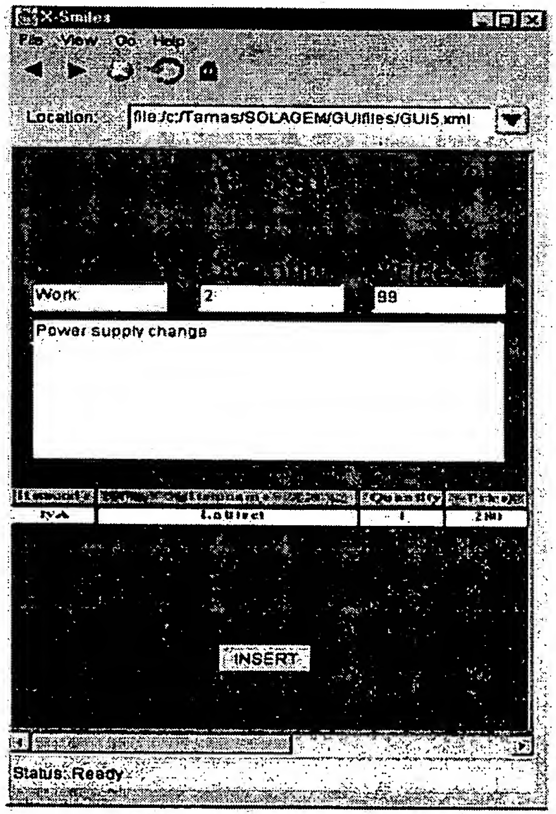


Figure 2. The motorcycle multimedia presentation, which is composed of images, text, and audio, is rendered using the SMIL MLFC.



file:/c:/Tamas/SOLAGEM/GUIfiles/GUI5.xml

Work

2

99

Power supply change

Status: Ready

Figure 3. A Sales order form, which contains data from Enterprise Resource Planning (ERP) system, is rendered using the XSL FO MLFC.

The XSL FO MLFC is based on the Apache XML FOP project (http://xml.apache.org/fop/). However, FOP cannot be used in the browser as such, as it provides only a way to preview a XSL FO document, but allows no means of user interaction – a very basic feature in any browser. Thus, we have extended the FOP to add user interaction using links and forms.

Simple links are defined in the XSL FO specification (i.e., the *fo:simple-link* element), while forms are not within the scope of the XSL FO. Forms are introduced in the XForms specification, which was during the implementation at an early stage. Thus, we used XHTML form elements (such as input fields, buttons, etc.) with the XForms data model [5] for rendering and input, and a subset of the XForms data model for defining the data model of the form. An example of a form rendered using the XSL FO MLFC is shown in Figure 3.

Finally, the SVG MLFC can render vector graphics documents. It uses a modified version of CSIRO SVG Toolkit (http://sis.cmis.csiro.au/svg/), which supports most of the SVG features. The SVG MLFC can be used to show vector graphics and the user can also manipulate them. There are still several things missing or only partially implemented, though. The reason for this is that the toolkit is not yet complete.

3.3 Graphical User Interfaces (GUIs)

The X-Smiles browser is intended for different kinds of consumer devices, as mentioned before. Different consumer devices have very different input and output devices. For example, the screen size varies a lot, e.g., from high-definition screen of digital television to small monochrome screens of mobile phones. Also, different kinds of input mechanisms are supported, e.g., remote control, touch screen with stylus, and buttons. In practice, these requirements mean that the GUI of the X-Smiles has to be versioned for different kinds of devices.

To support different kinds of consumer devices, the X-Smiles browser allows the use of different kinds of GUIs. The different GUIs are attached to the X-Smiles browser using so called bridge pattern [8], as shown in Figure 4. The *XSmilesUI* is the superclass of special GUIs. The special GUIs (i.e., *FtvGUI, NormalGUI,* and *LaptopGUI*) inhered the definitions from the XSmilesUI and are connected to the *Browser* via the *UIBridge*. The different GUIs can be swapped even on the fly.

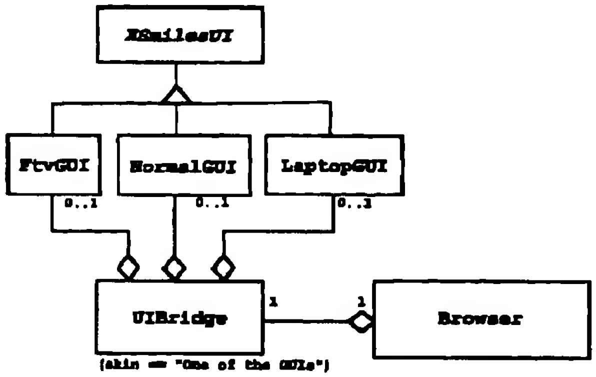


Figure 4. The GUI classes. The specialized GUIs inhered the properties of the XsmilesUI superclass.

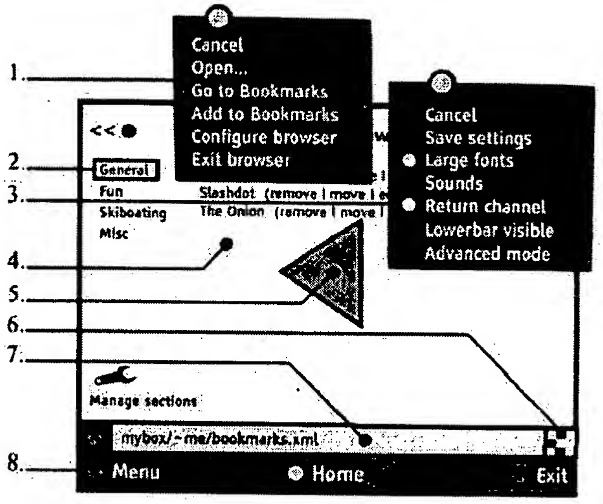


Figure 5. The digital television GUI: 1) main menu, 2) highlight, 3) configuration menu, 4) content area, 5) arrow, 6) animator, 7) statusbar, and 8) toolbar.

So far, we have implemented the above mentioned GUIs. The NormalGUI is the original GUI. As it was the original implementation, it was used primarily in the development environment (i.e., desktop PC). After that, we have implemented another GUI for laptop PCs. This GUI is visually better for demonstrations given from laptop PCs, as shown, e.g., in Figure 2. In addition, we have implemented special GUIs for PDAs and mobile phones.

Finally, we have also implemented a special GUI for digital television [18], as shown in Figure 5. This GUI has remote control as input device and television screen as output device. The focus point (i.e., highlight) of the GUI is controlled with the arrow buttons of the remote control. Selection is made with 'ok' button. The bottom of the screen contains the statusbar and the toolbar. The statusbar shows the current URL and has an animator for busy indicator. The toolbar options are activated by the color buttons of the remote control. The 'red' and 'green' buttons activate the main and configuration menus, respectively. The 'yellow' button selects the home page, while the 'blue' button is the exit.

3.4 Implementation

The X-Smiles browser was implemented using the Java language. Java is an optimum choice for two reasons. First, there are several public domain components available in Java as mentioned above. Second, the X-Smiles browser can be ported to different platforms, which have a Java Virtual Machine. Currently, the X-Smiles supports both Sun Microsystems JDK1.1 and JDK1.2. The current consumer devices have rather restricted Java support, and thus the JDK1.1 support is also important.

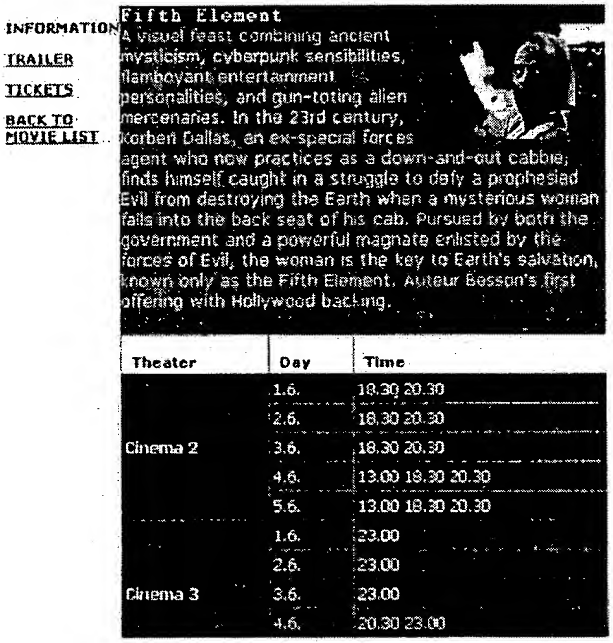
In addition to the basic JDK, some special APIs have to be used. Most important one of these is the Java Media Framework [10]. It is used primarily for continuous media, but also for displaying images, etc. The JMF takes care of streaming and rendering of continuos media. For streaming the Internet multimedia protocols are used [13].The GUI can be based either on Abstract Windowing Toolkit (AWT) or Swing. Since the user interface is separated in the GUI module, the windowing system can be easily changed. Swing gives more flexibility, but AWT is a better choice in limited environments, e.g., mobile devices.

In the development process, we have used various tools. For example, we used the Concurrent Versions System (CVS) (http://www.cvshome.org/) for version management and SourceForge (http://sourceforge.net/) for bug reporting.

4. RESULTS

The current X-Smiles version runs both on Windows and Linux platforms. The Linux platform has some complications due to the more limited JMF support, though. This situation should improve in the future, since better JMF implementations for Linux are under development.

The X-Smiles browser was tested with three trial services: movie theatre, distance education, and a multimedia presentation about tennis [16].



Fifth Element

A visual feast combining ancient mysticism, cyberpunk sensibilities, flamboyant entertainment personalities, and gun-toting alien mercenaries. In the 23rd century, Korben Dallas, an ex-special forces agent who now practices as a down-and-out cabbie; finds himself caught in a struggle to defy a prophesied Evil from destroying the Earth when a mysterious woman falls into the back seat of his cab. Pursued by both the government and a powerful magnate enlisted by the forces of Evil, the woman is the key to Earth's salvation, known only as the Fifth Element. Auteur Besson's first offering with Hollywood backing.

Theater	Day	Time
Cinema 2	1.6.	18.30 20.30
	2.6.	18.30 20.30
	3.6.	18.30 20.30
	4.6.	13.00 18.30 20.30
	5.6.	13.00 18.30 20.30
Cinema 3	1.6.	23.00
	2.6.	23.00
	3.6.	23.00
	4.6.	20.30 23.00

Figure 6. The movie theatre service contains information about available movies and theatres, and allows users to make ticket reservations.

The movie theatre service allows the user to view information about on-going movies, watch movie trailers, and make reservations for tickets. The service is intended for next generation mobile phones, as shown in Figure 6. The devices are supposed to have bigger color displays, but buttons are still used for input. Thus, the 'up' and 'down' buttons are used for traversing the menus. Selections are made with 'ok' button, while 'back' button can be used to return to the previous level of the menu structure. Number buttons are used for numerical input.

The distance education service allows the user to follow lectures as shown in Figure 7. Here, the supposed device is similar to the current PDAs, but has a bigger screen. Stylus and touch screen is used for input. The device should also have more processing power for multimedia. In this example, a lecture video is streamed over the wireless Ethernet network. The user can view the video and lecture slides simultaneously.

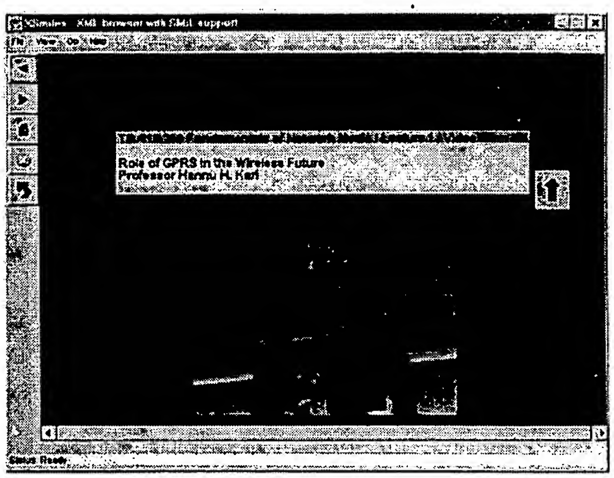
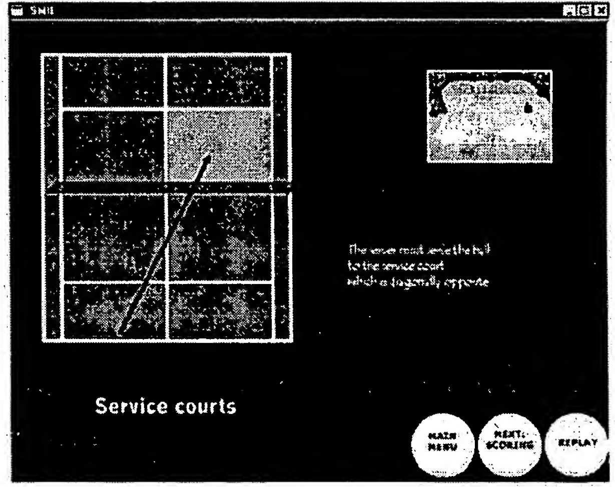

X-Smiles - XML browser with SMIL support
File
View
Go
Help
Role of GPRS in the Wireless Future
Professor Hannu H. Kari
Status Ready

Figure 7. The distance education service contains lecture video and slides.

In both of these services, the content is coded with XML, while the user interface is defined in a separate XSL stylesheet. The XSL stylesheet also contains the required ECMAScripts. The documents are rendered using XSL FO. The advantage of XSL is that the service can easily be fitted to different kinds of consumer devices by switching the XSL file.



The server must serve the ball to the service court which is diagonally opposite

Figure 8. The tennis service contains text, audio, images, graphics, and animations as synchronized multimedia presentation.

The two previous examples used their own set of XML tags, designed especially for the particular service. For more demanding multimedia applications SMIL should be used. As an example of such an application, we used the tennis service, which is composed of several media elements including, e.g., text, images, graphics, animations, and sound, as shown in Figure 8. The different elements are synchronized together into a multimedia presentation. The target device is similar to the current electronic books, but has a color display. The device should be powerful enough to display several multimedia elements simultaneously and synchronize them.

The above-mentioned services were originally developed for the Microsoft Internet Explorer 5.0 browser and GRiNS SMIL player [16]. After that, we have ported the applications to the X-Smiles browser, though. Especially, the services based on our own set of tags had to be modified to use XSL FO, because previously the XML data in these services was transformed into HTML for rendering. Currently, most SMIL presentations authored for the GRiNS player run in the X-Smiles with a few minor modifications.

The run-time environment has to be quite powerful, still. So far, emphasis has been on implementing features considered interesting from a service developer's point of view, and not much effort has been done to optimize the performance. For example, displaying SMIL presentations is not optimized, e.g., all media files are loaded into memory before the presentation starts (except when using streaming). This may result in high memory usage if the presentation contains many large media files. Using the Java Media Framework for all – even static – media elements is somewhat heavy and a better solution needs to be implemented. XML applications requiring less multimedia features can be easily shown in less powerful environments.

5. Conclusions

In this paper we have presented the X-Smiles browser, which we have been developing now for over three years. The browser is composed of several public domain modules and it is implemented using the Java language. This gives flexibility and allows the porting of the browser into different platforms.

Currently, the browser supports all the major XML specifications including, e.g., XSL Transformations, XSL Formatting Objects, SMIL, SVG, ECMAScript, and streaming of continuous media. There is a lot of work to be done, though. First, the current public domain components do not support all features. Thus, they have to be updated. Also, some XML related specifications of W3C have not been implemented at all (e.g., XHTML) or are supported at a minimum level (e.g., SMIL 2.0).

Currently, we are also working on some advanced, but interesting features. For example, we are integrating a Session Initialization Protocol (SIP) [13] client to the X-Smiles browser. Then, the browser can be used also as a videoconferencing tool.

Also, we are integrating a Java Message System (JMS) [11] client. The main advantage of JMS is that it can be used to push XML information to the browser. This is useful, for example, in eBusiness applications. An example of sales order form is depicted in Figure 3. Finally, we are also implementing specialized user interfaces for different kinds of mobile devices and porting the X-Smiles browser to real devices.

6. ACKNOWLEDGEMENT

The X-Smiles browser was originally developed in a student software project during 1998-99. After that, the development has continued in the GO-MM project. We would like to thank all the participants. The X-Smiles browser is available as open source at http://www.x-smiles.org.

7. REFERENCES

[1] Adler, S. et al. Extensible stylesheet language (XSL) version 1.0. W3C Working Draft, Oct. 18, 2000.

[2] Apparao, V. et al. Document object model (DOM) level 1 specification - version 1.0. W3C Recommendation, Oct. 1, 1998.

[3] Bosak, J. Media-independent publishing: four myths about XML. Computer 31, 10 (Oct. 1998), 120-122.

[4] Bray, T. et al. Extensible markup language (XML) 1.0 (2nd edition). W3C Recommendation, Oct. 6, 2000.

[5] Dubinko, M., Schnitzenbaumer, S., and Raggett, D. XForms 1.0: data model. W3C Working Draft, Aug. 15, 2000.

[6] ECMA-262. ECMAScript language specification. European Computer Manufacturers Association (ECMA), 1998.

[7] Ferraiolo, J. et al. Scalable vector graphics (SVG) 1.0 specification. W3C Candidate Recommendation, Aug. 2, 2000.

[8] Gamma, E., Helm, R., Johnson, R., and Vlissides, J. Design patterns: elements of reusable software. Addison-Wesley, 1994.

[9] Geier, J. Wireless LANs – Implementing interoperable networks. Macmillian Technical Publishing, 1998.

[10] Gordon, R. and Talley, S. Essential JMF: Java™ media framework. Prentice Hall, 1999.

[11] Hapner, M. et al. Java™ message system, ver. 1.0.2. Sun Microsystems, Nov. 9, 1999.

[12] Hoschka, P. et al. Synchronized multimedia integration language (SMIL) 1.0 specification. W3C Recommendation, June 15, 1998.

[13] Kosiur, D. IP Multicasting: The complete guide to interactive corporate networks. Wiley Computer Publishing, 1998.

[14] Kozyrakis, C. E. and Patterson, D. A. A new direction for computer architecture research. Computer 31, 11 (Nov. 1998), 24-32.

[15] Lie, H. W. and Saarela, J. Multipurpose web publishing using HTML, XML, and CSS. Communications of the ACM, 42, 10 (Oct. 1999), 95-101.

[16] Marttila, O. and Vuorimaa, P. XML based mobile services, in Proc. WSCG'2000 8th Int. Conf. in Central Europe on Computer Graphics, Visualization, and Interactive Digital Media (Plzen, Czech Republic, Feb. 7-10, 2000).

[17] Sivaraman, G. and Vuorimaa, P. Compact windowing system for mobile devices, in Proc. MMSA2000 2nd Int. Symposium on Mobile Multimedia Systems & Applications (Delft, The Netherlands, Nov. 9-10, 2000), 134-141.

[18] Vierinen, J. and Vuorimaa, P. A browser user interface for digital television, in Proc. WSCG'2001 9th Int. Conf. in Central Europe on Computer Graphics, Visualization and Computer Vision (Plzen, Czech Republic, Feb. 5 - 9, 2001), 174-181.

[19] Vuorimaa, P. A XML based mobile software architecture, in Proc. MMSA2000 2nd Int. Symposium on Mobile Multimedia Systems & Applications (Delft, The Netherlands, Nov. 9-10, 2000), 150-156.